# Tutorial 10: Uninformed search, informed search, adversarial search

## Uniformed search

1. Summarize the advantages and disadvantages of depth-first search and breadth-first search.

|  |  |  |
| --- | --- | --- |
|  | advantage | disadvantage |
| Depth first search | * The space complexity is good; it is branch\*(depth of the whole tree) | * Not optimal * not guarantee to get the goal, if the depth is infinite. * The time complexity is: branch^(depth of the tree) |
| Breadth first search | * Can get optimal result * guarantee to get the goal | * The space complexity is branch^(depth+1). * The time complexity is branch^(depth+1). |

1. How does the iterative deepening search obtain the advantage of both methods while avoiding their disadvantage?

Iterative deepening search is an algorithm which do the depth first search with a certain depth limit, if in the process it doesn’t find the goal, it will increase the limit depth and start from the beginning to do it again; it keeps increase the depth limit until it finds the goal.

During the whole process, the iterative deepening search can get optimal result, and guarantee to get the goal, it also avoids the breadth first search’ disadvantage which its time complexity is in worst case: branch^depth, much fewer than branch^(depth+1). It also has the depth first advantage on space complexity, it is branch^ (limit of depth).

Informed Search

1. How do informed search differ from uninformed search?

Uninformed search just systematically brute search the state space which consist of every possible situation. Thus, it is not efficient.

But during informed search, because we know some information about the problem, so we can improve or be smart when we organize how the nodes added to the fringe. The basic idea is that put nodes which are ‘likely to lead to a good solution’ towards the front.

1. What is best first search?

When we put nodes on fringe, we define an evaluation function to measure it. For conventions, the lower value means better. So, all the nodes in fringe are kept in order of increasing. So every time we process the fringe, we can always start from the best possible one.

1. What is a heuristic?

Heuristic is a way of measuring how good a state is likely to lead to a solution. The principles we used to define heuristic are often the “rules of thumb” or some rules that works most of the time.

1. What is greedy search?

Greedy search is a way of informed search which during this process, we simply choose the shortest path to goal while regardless the previous cost of being current situation. So the evaluation function is just the heuristic function.

1. Descript A\* search algorithm. Why it is better than greedy search?

Because it counted the cost of being through previous states. It makes it can find the goal, it is complete and it is also optimal (if the heuristic function is admissible).

1. Fringe:

* (Copernicus [0+8=8])
* (Reinold [3+4=7], Eratosthenes [2+6=8])
* (Eratosthenes [2+6=8], Lansberg [3+2+5=10], Copernicus [3+8=11])
* (Trieshecher [2+6+0=8], Lansberg [3+2+5=10], Copernicus [3+8=11], Copernicus [2+2+8=12])
* Trieshecher is the goal, done

1. What is admissible heuristic? Explain informally how an admissible heuristic guarantees A\* search is optimal.

Admissible heuristic is one that never overestimates the cost to reach to goal.

Use straight distance as admissible heuristic:

Because g(n) is the actual cost to reach n along the current path, and f(n) = g(n) + h(n), we can get that f(n) will never overestimates the true cost of a solution along the current path through n. By nature, admissible heuristic is optimistic because it assumes the cost of solving the problem is less than its actually cost.

## Adversarial Search

1. Describe the Minimax algorithm for search in two-player games.

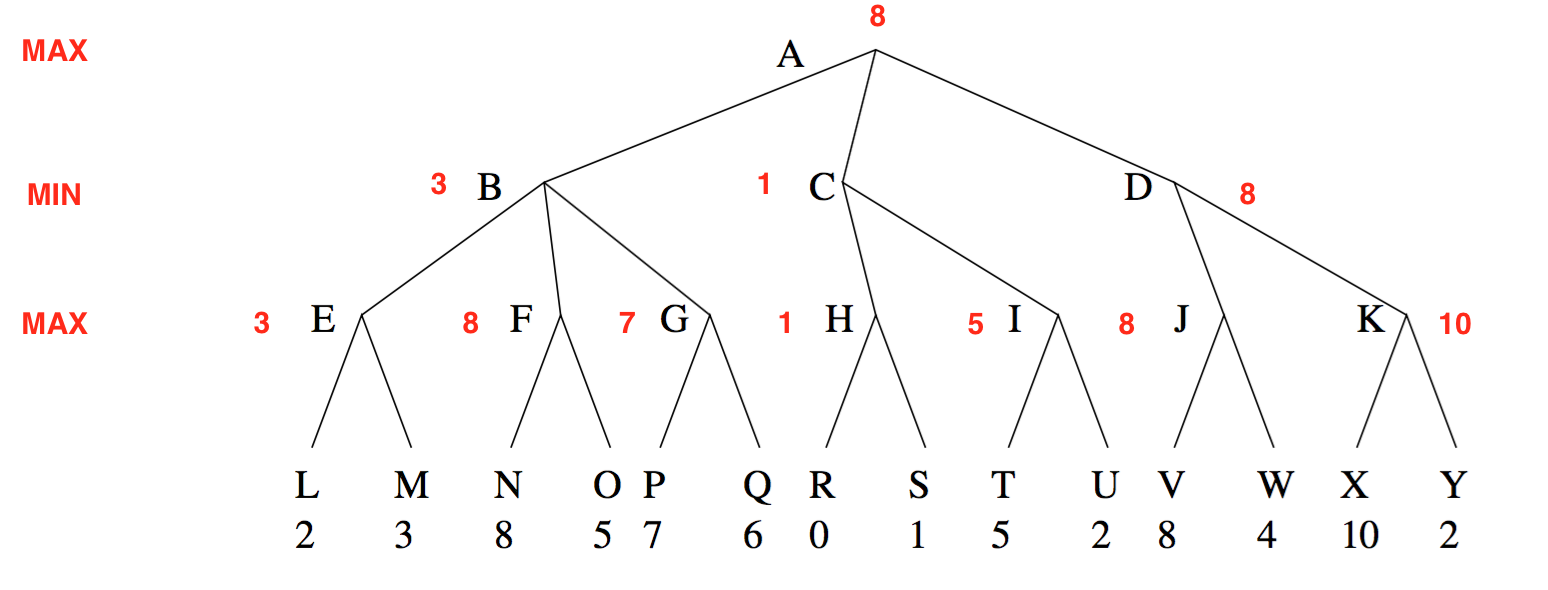
Suppose it is Max’s move at beginning:

1. Max takes the move that have the max value among possible move;
2. Each of these max value comes from the minimum value among each associated Min’s move.

These a), b) happened alternatively down to a state of ending game, then the utility value back up to the top.

It computes the Minimax decision from the current state. It uses recursive computation of Minimax values of each successor state. The recursion proceeds all the way down to the leaves of the tree, and then the Minimax values are backed up through the tree as the recursion unwinds.

1. At the end of the game it is Max’s move, so it chooses the max value, back up then:



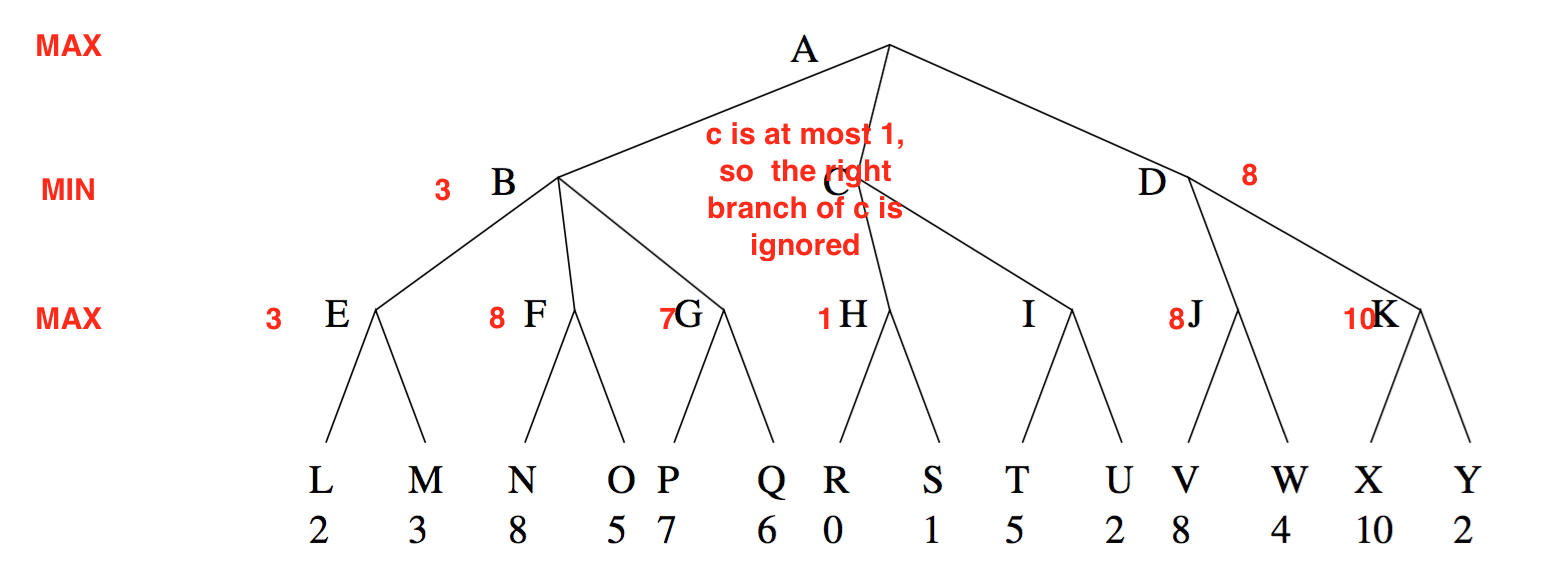
1. Use Minimax to determine which move that the first player should choose:

So, the first player should choose D.

1. What nodes would not need to be examined using the alpha-beta pruning algorithm,

Assuming that nodes are examined in left to right order.

From depth-first search:



Because B=3, and C’s value is at most 1, so the right branch of C, which include I, T and U is ignored.